Online Retail SalesDatabase Design

Introduction

In the current digital era, efficient management of e-commerce data is crucial for the growth and sustainability of online retail businesses. This project delivers a relational database design suitable for managing an online retail store, covering products, customers, orders, payments, and associated sales reports. The system ensures both data integrity and simplified query capabilities for operational and analytical purposes.

Abstract

The goal of this project is to build a robust and scalable database that encapsulates the essential components of an online retail system. The system includes tables to manage products, customers, orders, payments, and order items. Relational integrity is maintained using appropriate constraints and foreign keys. The database allows for comprehensive sales reporting using SQL JOIN queries and database views. An Entity-Relationship (ER) diagram, structured SQL schema, sample data, and analytic queries demonstrate the database's functionality and analytics capabilities.

Tools Used

- dbdiagram.io: Used to visualize the database design and generate the ER diagram.
- PostgreSQL: Relational database management system used for schema creation, data insertion, and query execution.
- SQL: Structured Query Language used for defining the schema, inserting sample data, and extracting meaningful reports.

Steps Involved in Building the Project

1. Requirements Analysis: Key entities (Products, Customers, Orders, Payments, Order Items) and their relationships were identified based on standard retail operations.
2. ER Diagram Design: Using dbdiagram.io, an ER diagram was created to illustrate the relationships:

- Products (One-to-Many with Order Items)
- Customers (One-to-Many with Orders)
- Orders (One-to-Many with Order Items & Payments)
- Payments (One-to-One/Many with Orders)
- Order Items (Associative entity between Orders and Products)

1. Schema Definition: SQL Data Definition Language (DDL) scripts were written for table creation, ensuring normalization and referential integrity:

- Use of `SERIAL PRIMARY KEY`
- `NOT NULL`, `UNIQUE`, and `CHECK` constraints

- Foreign keys with `ON DELETE CASCADE` for related deletions

1. Sample Data Insertion: Representative records were inserted into each table to simulate real-world operations (e.g., customers placing orders and making payments for products).
2. Querying the Database:

- A comprehensive JOIN query was implemented to fetch order details, including customer information, ordered products, and payment method.
- A database view (`sales_report`) was created to provide summarized sales insights (total sales per order, customer info, item count).

1. Validation & Testing: Queries were executed to test relationships, integrity constraints, and reporting capabilities.

Conclusion

This project demonstrates an efficient and normalized database solution for an online retail sales system. The schema captures all major entities and their relationships central to e-commerce workflows. By implementing analytical queries and reporting views, the database supports both transaction processing and business intelligence needs. Furthermore, the design is scalable: it can accommodate additional features such as product categories, shipment tracking, and user authentication, making it suitable for real-world deployment in e-commerce scenarios.